In the Claims:

1. A method for dynamically generating a wrapper object, comprising:

receiving a vendor object and superclass;

performing reflection on the vendor class;

generating a wrapper class;

instantiating the wrapper class, the instantiating including generating a wrapper object as an instance of the wrapper class; and

associating the vendor object with the wrapper object.

2. The method of claim 1 wherein the wrapper object is dynamically generated at runtime.

3. The method of claim 1 wherein the superclass is one of a pre-existing JDBC, JMS, or connector class.

4. The method of claim 1 wherein the superclass includes logic to handle server side tasks.

5. The method of claim 1 wherein the wrapper class is generated in bytecode.

6. The method of claim 5 wherein bytecode is generated for vendor methods not implemented in the superclass.

7. The method of claim 5 wherein the bytecode is generated using hot code generation.

8. The method of claim 1 wherein providing the wrapper object to the an application program, allowing the application program to access standard features and non-standard vendor extensions.

9. The method of claim 1 wherein the standard extensions are J2EE extensions.

10. A method for processing an invocation using a dynamically generated wrapper, comprising:

receiving an invocation call by a wrapper object, the invocation call directed to a wrapped vendor object by an application program;

initiating pre-processing by the wrapper object;

calling the wrapped vendor object by the wrapper object;

receiving a result from the wrapped vendor object by the wrapper object;

initiating post-processing by the wrapper object; and

provide the result to the application program.

11. The method of claim 10 wherein the pre-processing including calling a pre-invocation handler.

12. The method of claim 10 wherein the pre-invocation handler is configured to execute server-side code.

13. The method of claim 12 wherein the server-side code includes global transaction

processing code.

14. The method of claim 10 wherein post-processing including calling a post-invocation handler.

15. The method of claim 14 wherein the post-invocation handler is configured to perform post-processing server side tasks.

16. The method of claim 15 wherein the post-processing server-side tasks include global transaction management.